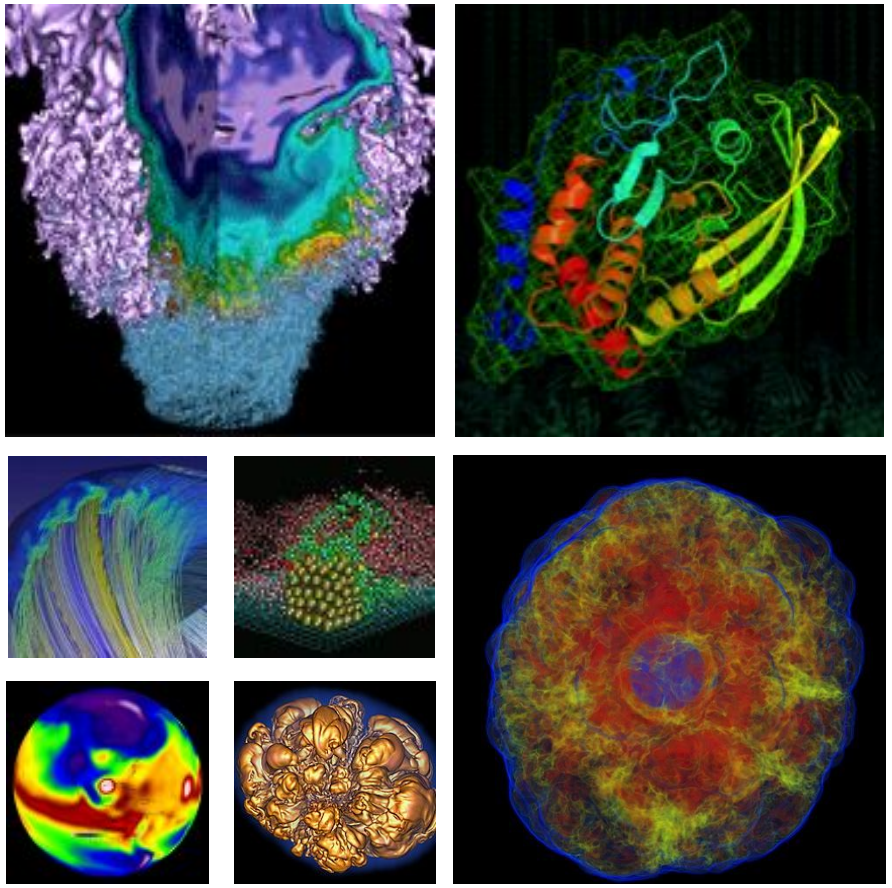
NUG Monthly Telecon



NERSC

12th May 2016

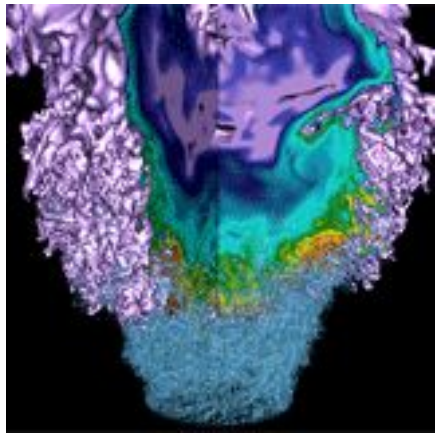
U.S. DEPARTMENT OF ENERGY | Office of Science

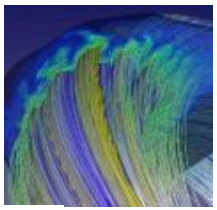
BERKELEY LAB
Lawrence Berkeley National Laboratory

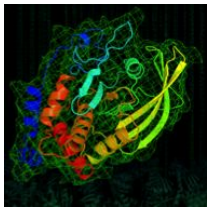
Agenda

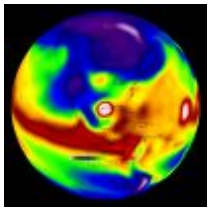
- System updates
- Move/outage update, Rhine Redwood OS upgrade
- Accounts and charging corrections
- File system licenses
- HPC software engineering seminar series
- The NERSC Burst Buffer: Early User Experiences

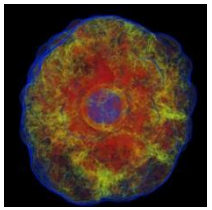
Cori Update

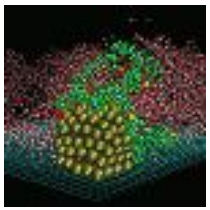












Richard Gerber

NUG Monthly, 5/12/2016

Upcoming Cori Outages

- **NERSC Quarterly Maintenance**
  - Tuesday, May 24, 7 a.m.-8 p.m. PDT
- **Upgrade to Cray Linux Environment (CLE) 6.0 (Rhine/Redwood)**
  - Cori will be offline starting June 6 for up to two weeks
  - CLE 6.0 is required to support integration with the Cori Phase 2 KNL system and contains significant system management enhancements
  - Will require you to rebuild applications
  - NERSC will be working to return the system as soon as possible
- **Cori Phase 1 and 2 Integration**
  - Approximately September 2016
  - 4-6 weeks of downtime on Phase 1

1 node jobs can now run for 96 hours

- **Per user run limit: 4 jobs**
- **Per user submit limit: 10 jobs**
- **Global run limit: 10 jobs**
- **Regular charge only; premium and low" not available**
- **Usage example:**

  % sbatch -t 96:00:00 -p regular -N 1 myscript
- **Note**
  - Jobs can not start if their requested wallclock exceeds the time until the next scheduled maintenance

Beta Cray Compilers

- **You can help test and debug the next Cray development environment release by trying Cray Compiling Environment (CCE) 8.5.0 Beta.**
- **To use on Cori or Edison**
  - % module swap PrgEnv-intel PrgEnv-cray
  - % module swap cce cce/8.5.0.4664
- **Key enhancements**
  - Support for the C11 language standard
  - Performance improvements
- **Please send us comments and bug reports**
  - Email consult@nersc.gov

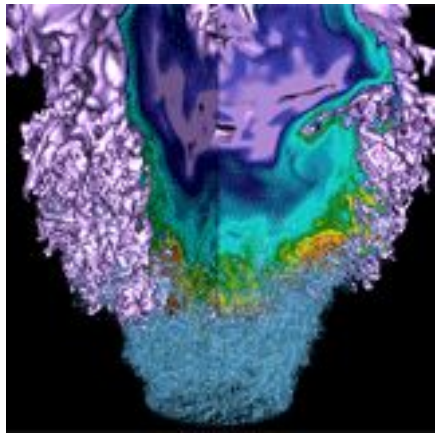
Cori Phase 2 Planned Timeline

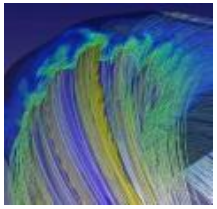
- **May to Sept 2016**
  - KNL white boxes available for NESAP teams
- **Mid July to August 2016**
  - Phase 2 cabinets arrive
  - System configuration and testing
- **Late August to September 2016**
  - Phase 1 and Phase 2 integration and initial acceptance testing
  - 4-6 weeks of outage on Phase 1
  - We are discussing how to support data science workloads on Edison
- **September 2016**
  - Cori acceptance testing starts
- **NESAP early user period starts when system is ready**
- **See** https://www.nersc.gov/users/cori-phase-ii-schedule/

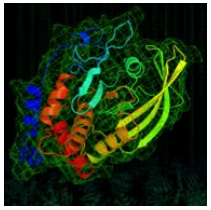
Charging and allocation usage

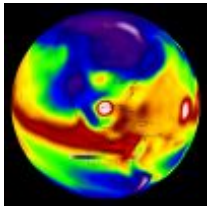
- **We will be recalculating some account charges.**
  - We were missing jobs that ran over consecutive midnights. We updated April already and will fix January through March in the coming weeks.
  - Full reservation charges will be attributed to accounts in the coming weeks.
- **Reminder about scavenger queue when you run out of time**
  - Submit jobs as normal and they will run when resources become available
  - 32 repos are currently negative
  - 12 M hours used in scavenger this year so far

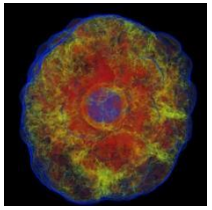
File System Licenses

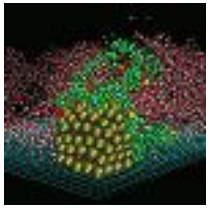












File System Licenses

- **Users can now specify the file systems required for their jobs by requesting licenses in their batch scripts**
- **Jobs that uses file system licenses will not start if there is an issue with that particular file system**
  - Protects jobs from failing if there's an issue with that file system
  - Allows selective maintenance on an individual file system

How to Use File System Licenses

- **File system licenses can be used in SBATCH lines in your batch script or as command line flags**

```
#SBATCH –L scratch1

#SBATCH –L scratch1,project

#SBATCH --license=scratch1
```

- **Can also use SCRATCH (all caps) as short hand for your scratch directory (i.e. /scratch1 or /scratch2 on Edison or /global/cscratch1 on Cori)**

```
#SBATCH –L SCRATCH,project
```

List of File System Licenses

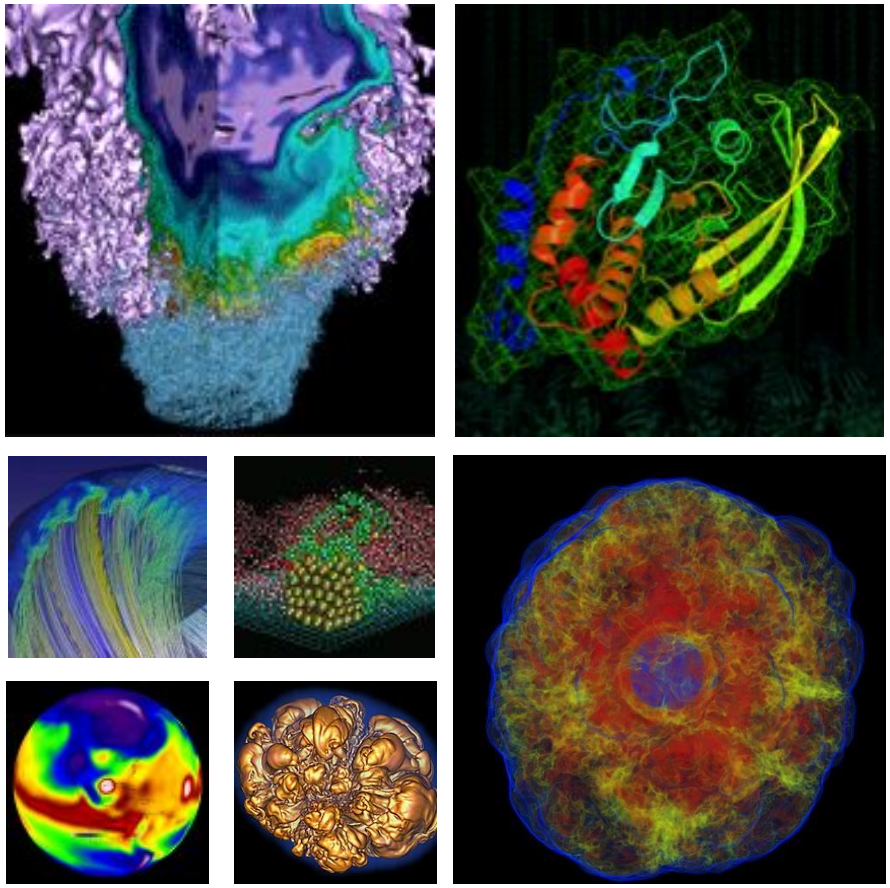
Computational System	File System	License
Cori	/global/cscratch1	cscratch1 or SCRATCH
Cori	/global/project	project
Cori	/global/projecta	projecta
Cori	/global/projectb	projectb
Cori	/global/dna	dna
Edison	/scratch1	scratch1 or SCRATCH
Edison	/scratch2	scratch2 or SCRATCH
Edison	/scratch3	scratch3
Edison	/global/project	project
Edison	/global/projecta	projecta
Edison	/global/projectb	projectb
Edison	/global/dna	dna

Homes and common are not licensed because they are essential for all jobs

Please Use File System Licenses

- **We strongly encourage you to start using file system licenses in all your jobs**
  - Benefits you by protecting your job from file system issues
  - Helps NERSC to better manage and plan for file system down times
- **Use of licenses is voluntary for now, but may become mandatory in the future**
- **Please contact consult@nersc.gov if you encounter difficulties using file system licenses**

HPC Software Seminar Series



NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

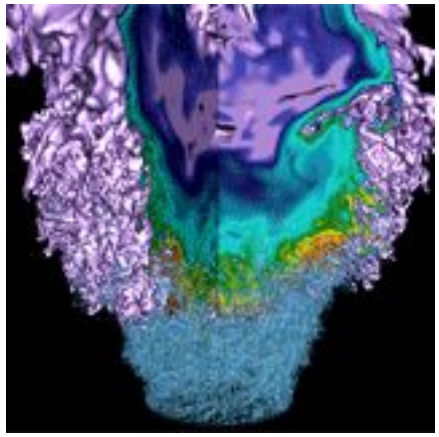
Best Practices for HPC Software Developers Webinar Series

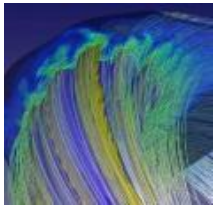
- **Joint project with IDEAS Software Productivity Project, ALCF, NERSC, OLCF**
- **Series of seminars bi-weekly, May through beginning of August**
- **Participation in prior sessions not required**
- **Topics covered include:**
  - Best practices in software development
  - Version control and continuous integration testing
  - Testing and documenting code
  - Performance analysis and code optimization
  - I/O on HPC systems

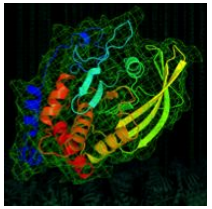
Best Practices for HPC Software Developers Webinar Series

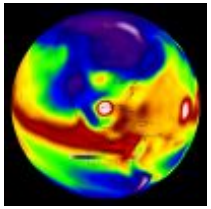
- **Next webinar: Wednesday, May 18, 10:00 am PDT**
  - Barry Smith (ANL) presents "Developing, Configuring, Building, & Deploying HPC Software"
- **Past presentation videos available in playlist on NERSC Training YouTube channel: https://www.youtube.com/playlist?list=PL20S5EeApOSuE6BZWEbMMV-9-ZHNAu3NI**
- **To register or for more information: https://www.olcf.ornl.gov/training-event/webinar-series-best-practices-for-hpc-software-developers/**

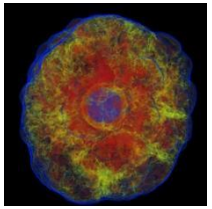
Burst Buffer Early User Program

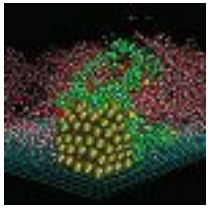












Outline

- **Why a Burst Buffer?**
  - Architecture and software
- **Users are excited about new architectures!**
  - Early User Program
- **Science applications ≠ benchmarks**
  - Real-world performance
- **New tech teething problems**
  - Challenges and Lessons Learned

HPC memory hierarchy is changing

Past (Edison)

On Chip

CPU

Memory (DRAM)

Off Chip

Storage (HDD)

Future (Cori)

On Chip

CPU

Near Memory (HBM)

Far Memory (DRAM)

Off Chip

Near Storage (SSD)

Far Storage (HDD)

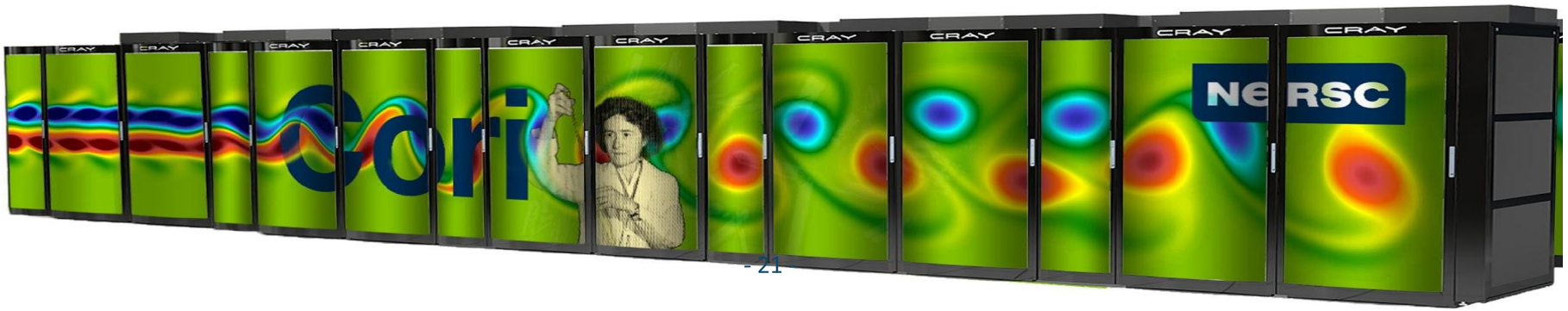
Why a Burst Buffer?

- **Motivation: Handle spikes in I/O bandwidth requirements**
  - Reduce overall application run time
  - Compute resources are idle during I/O bursts
- **Some user applications have challenging I/O patterns**
  - High IOPs, random reads, different concurrency…
- **Cost rationale: Disk-based PFS bandwidth is expensive**
  - Disk capacity is relatively cheap
  - SSD *bandwidth* is relatively cheap

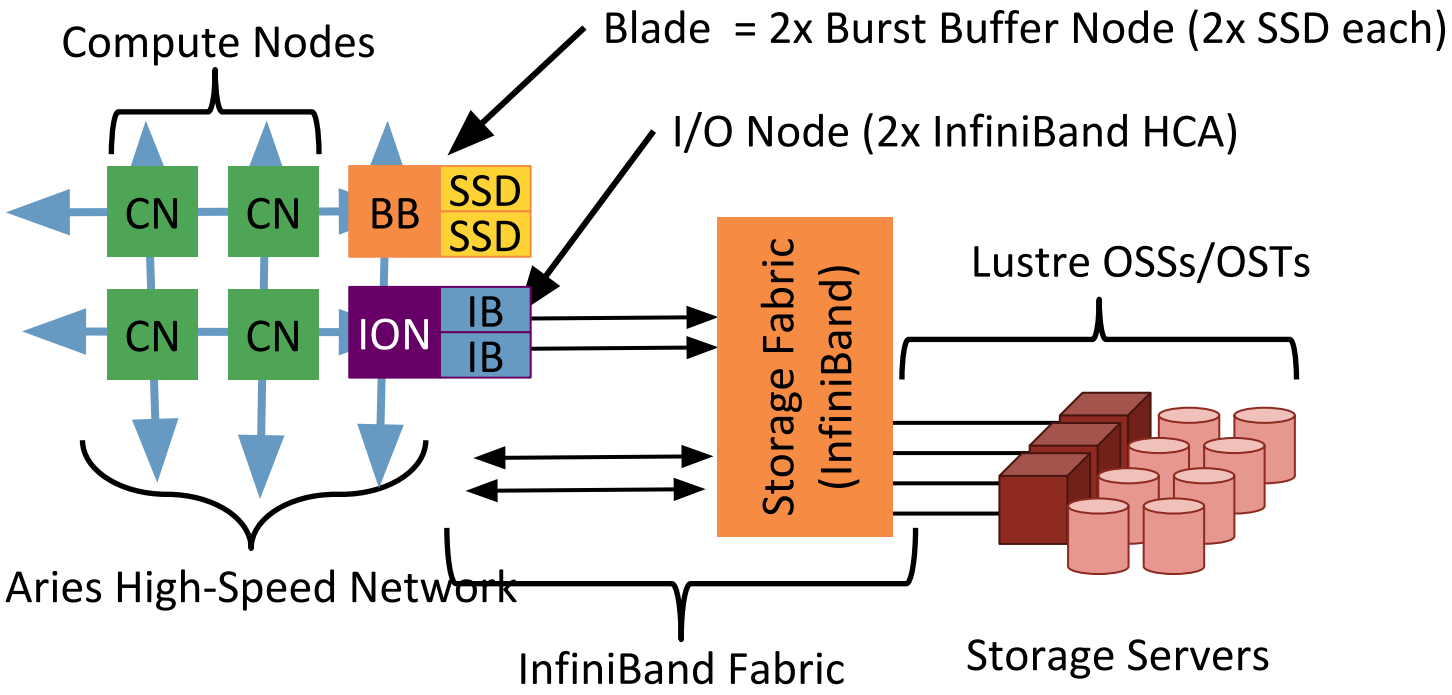
    =>Separate bandwidth and spinning disk
    - Provide high BW without wasting PFS capacity

Cori, a Cray XC40 system

- **Cori Phase 1: partition to support data intensive applications**
  - 1630 Intel Haswell nodes
  - Two Haswell processors/node,
    - 16 cores/processor, 128 GB DDR4 /node
- **Cori Phase 2: >9,300 Intel Knights Landing compute nodes**
  - 16GB HBM on-package, 96GB DDR4
- **Lustre Filesystem: 27 PB of storage served by 248 OSTs, providing over 700 GB/s peak performance.**
- **Cray Aries high-speed "dragonfly" topology interconnect**
- **1.5PB Burst Buffer…**



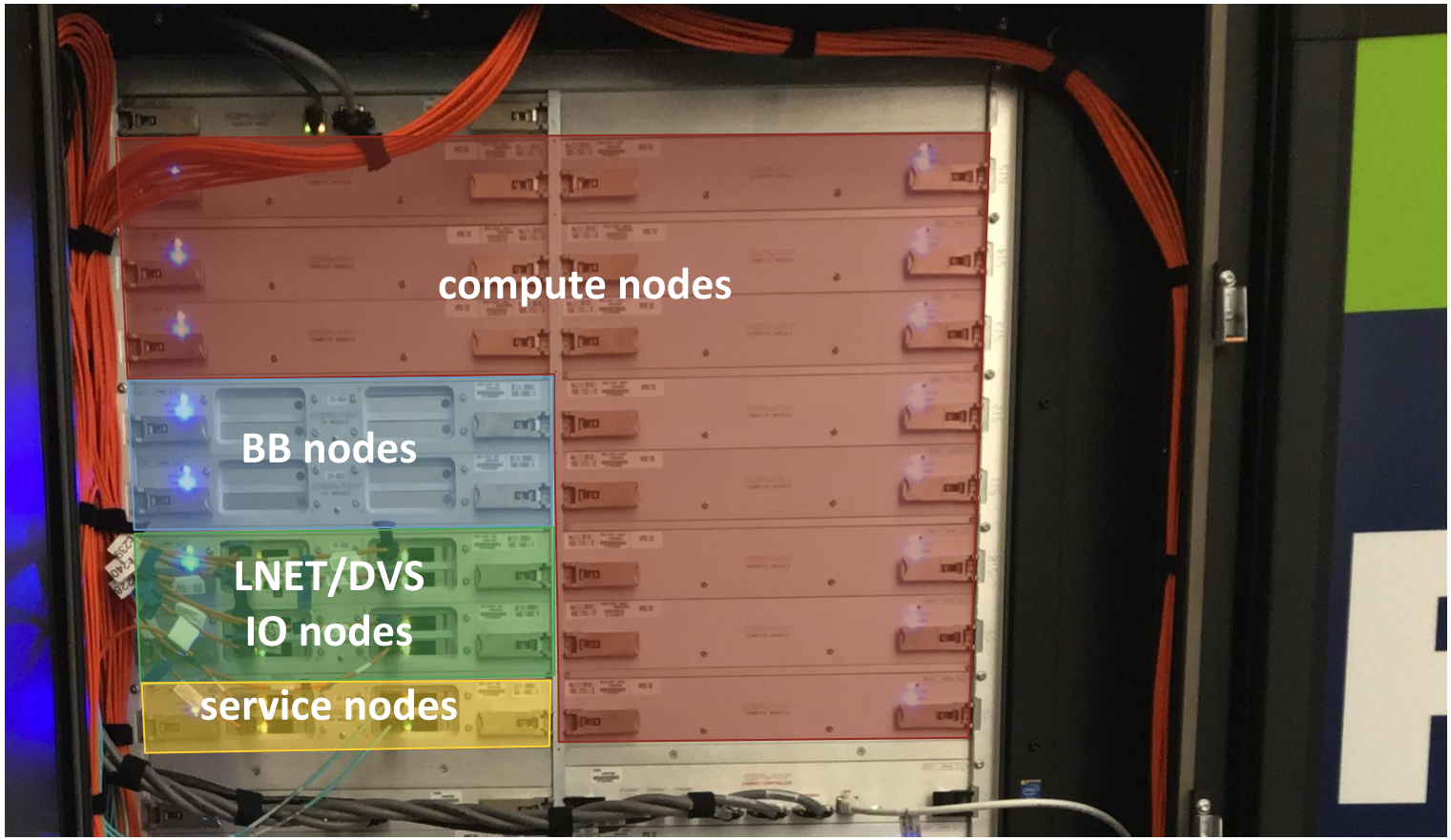
Burst Buffer Architecture



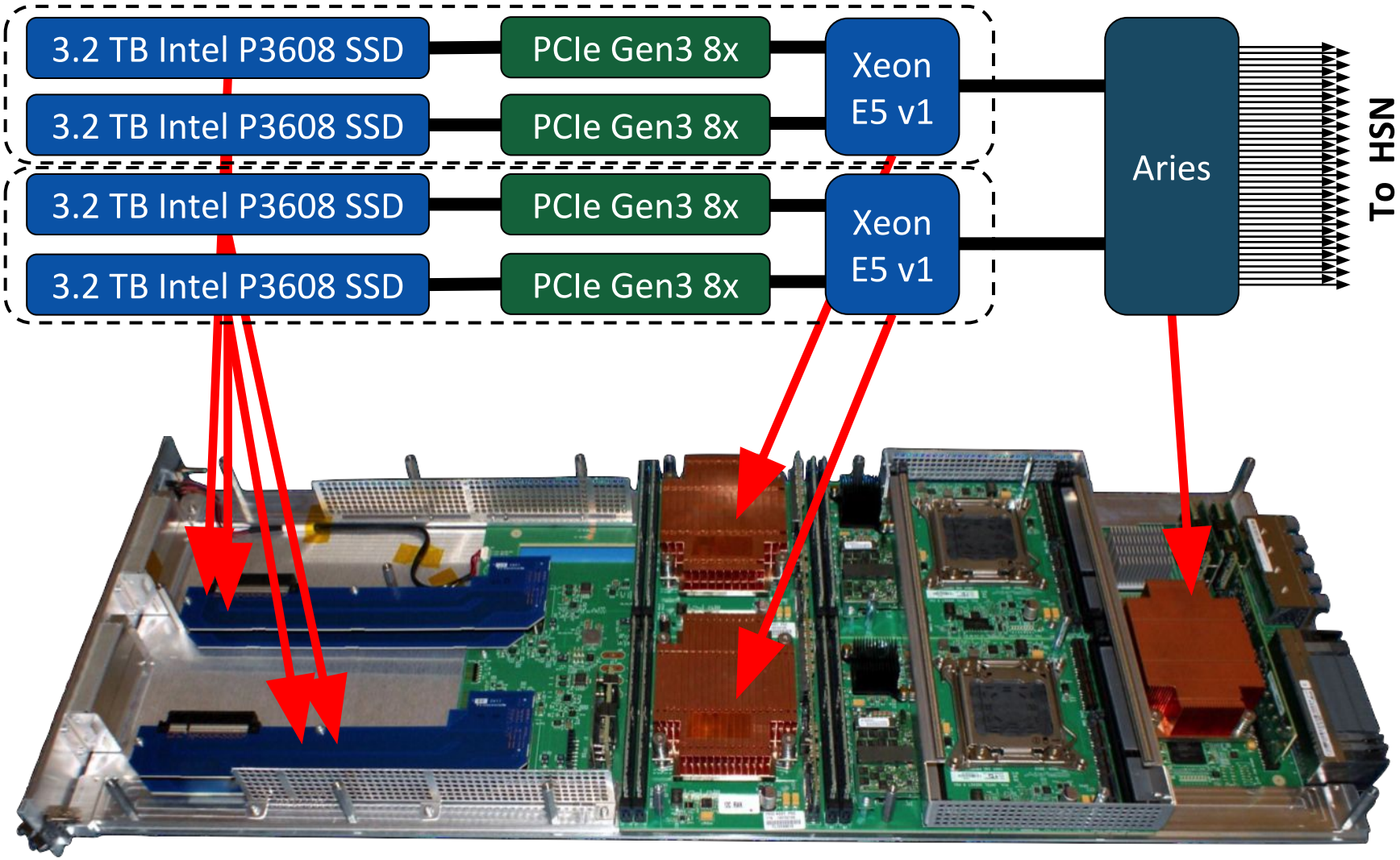
- Cori Stage 1 configuration: 920TB on 144 BB nodes (288 x 3.2 GB SSDs)
- >1.5 PB total coming with Cori Phase 2

Burst Buffer Architecture Reality

BB nodes scattered throughout HSN fabric
2 BB blades/chassis (12 nodes/cabinet) in Phase I



Burst Buffer Blade = 2xNodes



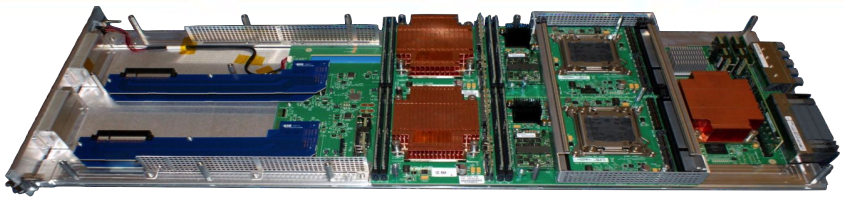
Why not node-local SSDs?

- **Average >1000 jobs running on Cori at any time**
- **Diverse workload**
  - Many NERSC users are IO-bound
  - Small-scale compute jobs, large-scale IO needs
- **Persistent BB reservations enable medium-term data access without tying up compute nodes**
  - Multi-stage workflows with differing concurrencies can simultaneously access files on BB.
- **Easier to stream data directly into BB from external experiment**
- ***Configurable BB makes sense for our user load***

Cray DataWarp software layers

- **Logical Volume Manger (LVM)** groups the SSDs into one block device.
- An **XFS** file system is created for every Burst Buffer allocation
  - Per-job “scratch”, or persistent reservation.
- **DataWarp File System (DWFS)**: stacked file system providing the namespaces.
- **Cray Data Virtualization Service (DVS)**: mediates communication between DWFS and the compute nodes.

Software developed in NRE with Cray



Burst Buffer Integration

- **Resource manager**
  - Good integration between SLURM/DataWarp
    - Stages data in/out before/after job hits the compute nodes
    - Keeps track of SSD wear and tear, allocates accordingly
    - [however, currently do not co-schedule BB and compute]
    - API can asynchronously bleed data off SSDs
- **Flexible memory management**
  - Data can be visible to compute node only, to all compute nodes, or all jobs.
  - User must define data movement (pros and cons)
  - [Transparent caching coming soon (pros and cons)]
- **MPI**
  - SSDs on service nodes inside Aries network – avoid contention if want distributed FS across multiple BB nodes.

Benchmark Performance

- **Burst Buffer is exceeding (nearly all) benchmark performance targets**
  - Work on-going to improve MPIO shared file write
  - Out-performs Lustre (Lustre also exceeds requirements)

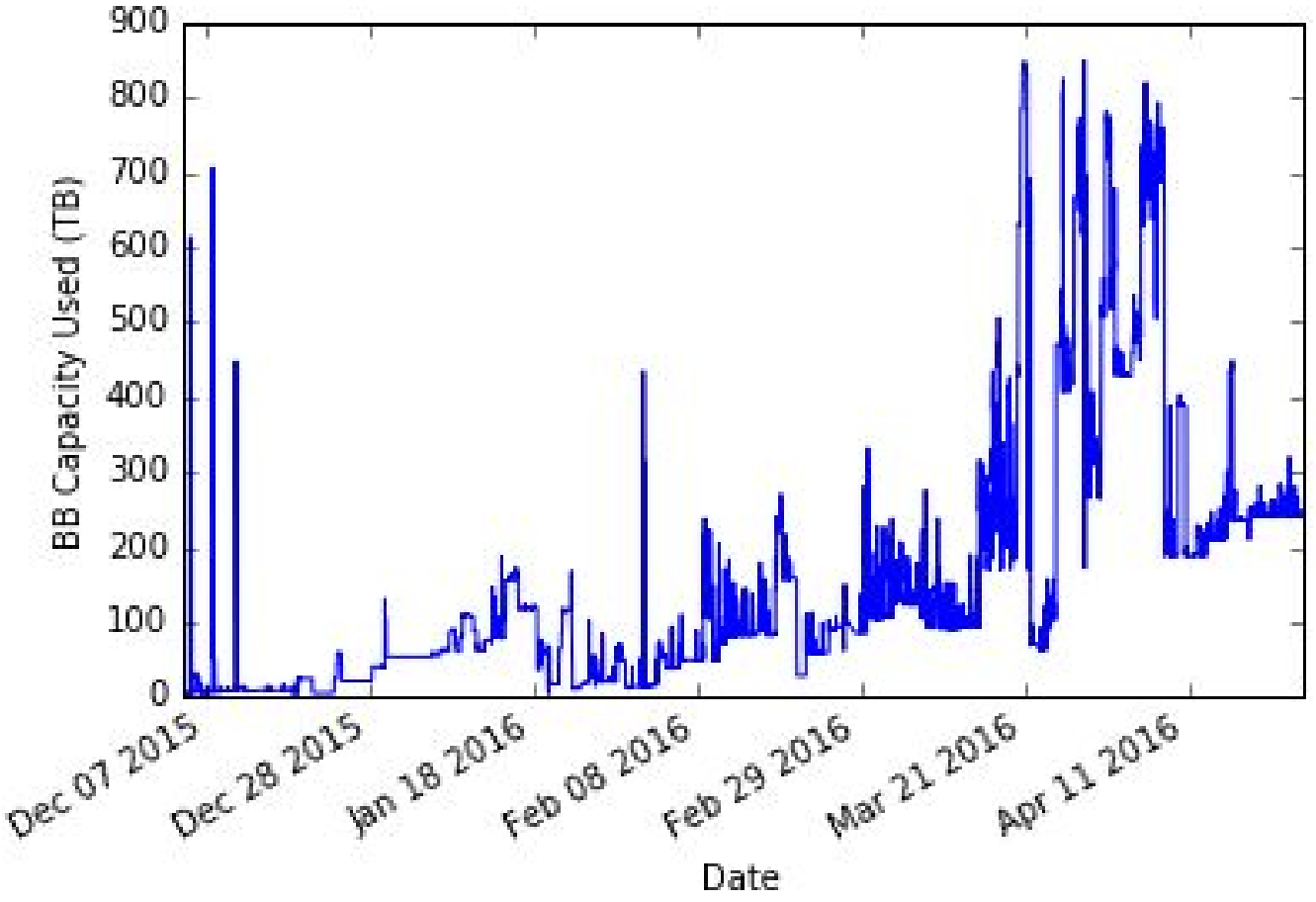
	140 Burst Buffer Nodes : 1120 Compute Nodes; 4 processes/node					
	IOR Posix FPP		IOR MPIO Shared File		IOPS	
	Read	Write	Read	Write	Read	Write
Best Measured	905 GB/s	873 GB/s	803 GB/s	351 GB/s	12.6 M	12.5 M
Lustre (peak)	708 GB/s	751 GB/s	573 GB/s	223GB/s	-	-

*Bandwidth tests: *8 GB block-size 1MB transfers*
IOPS tests: 1M blocks 4k transfer

Burst Buffer Early User Program

- **NERSC has most diverse user base of all DOE computing facilities: Over 6500 users on more than 700 projects, running over 700 codes**
- August: solicited proposals for BB Early Users program.
- Great interest from the community, ~30 proposals received.
- Selection criteria include:
  - Scientific merit; Computational challenges; Cover range of BB data features; Cover range of DoE Science Offices.
- Support ~10 applications actively
  - some applications already had LDRD funding at LBNL, and existing support from NERSC staff.
- ~20 applications not supported by NERSC staff, but have early access to Cori P1 and the BB.

Burst Buffer Occupation



~50 active users, not general access

User Experience

- **Significant number of major software bugs continue to impact user experience**
  - Most have been quickly patched by Cray
- **Minor bugs/quirks cause some frustrations**
  - E.g. formatting requirements, soft links on the BB...
  - Also quickly patched by Cray
- **Few users saw OOTB improvement in IO**
  - Most saw (see) far better performance on Lustre
  - Significant effort required to get good performance out of existing code

Use-cases with results

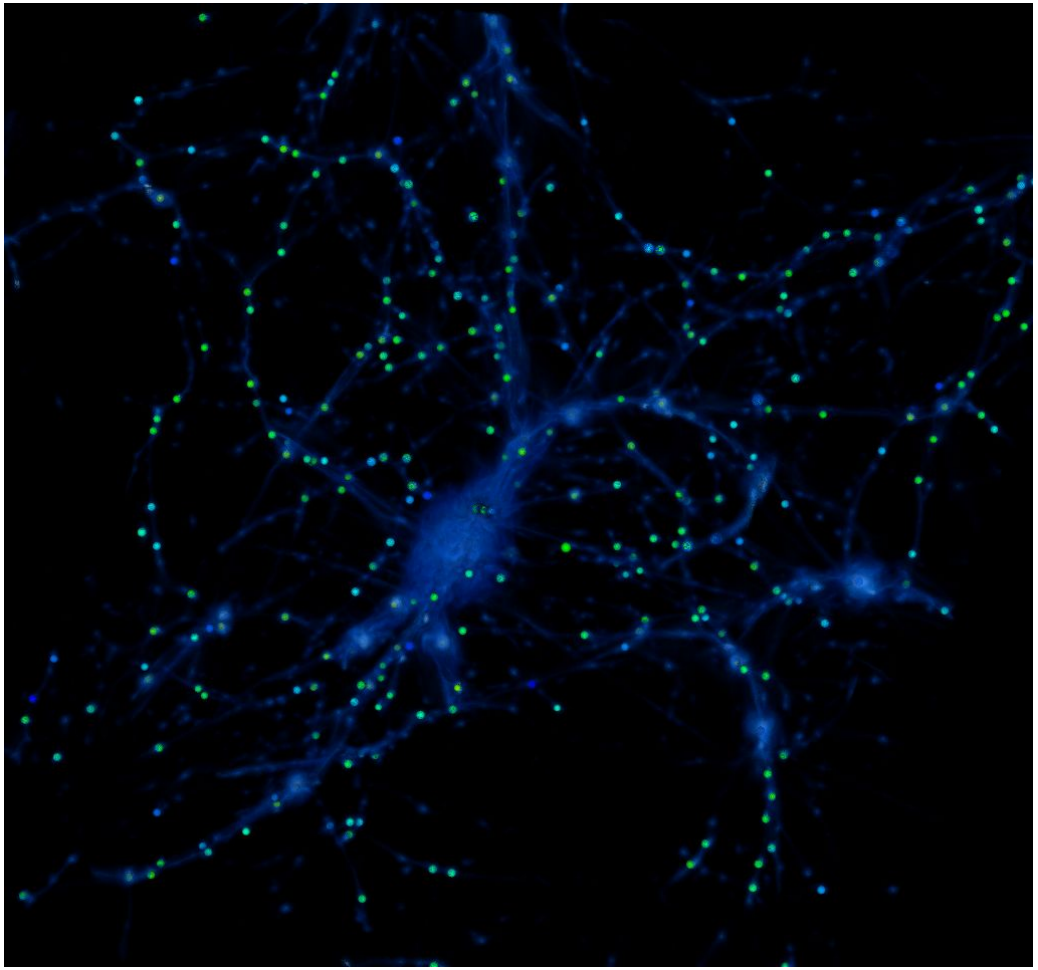
Burst Buffer User Case	Example Early Users
IO Bandwidth: Reads/ Writes	• **Nyx/BoxLib astro sims** • VPIC IO plasma sims
Data-intensive Experimental Science - “Challenging” IO pattern, eg. high IOPs	• **ATLAS HEP experiment** • TomoPy for ALS and APS • Genome assembly codes
Workflow coupling and visualization: in transit / in-situ analysis	• **ChomboCrunch & VisIt carbon sequestration simulation** • Climate simulation/visualization • Electron cryo-microscopy image assembly/visualization
Staging experimental data	• **ATLAS HEP experiment** • ALS SPOT Suite • Tractor astronomy image analysis

I/O Performance: Read/Write

- **Classic "checkpoint" use case also applies to our data-intensive users writing out large simulation data files**
- **To maximise BB BW, we need to keep it busy:**
  - Need several processes writing to a BB node
  - Need large transfer sizes
- **Use cases that fit this I/O pattern (or can adapt to it) saw good performance compared to Lustre**

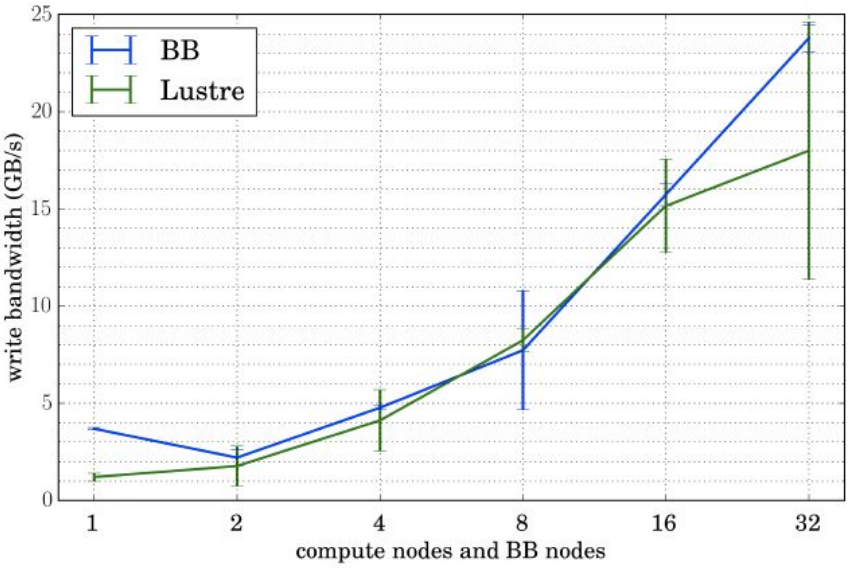
I/O R/W use case: Nyx/Boxlib

- Nyx cosmological simulation code based on a widely-used adaptive mesh refinement (AMR) library, BoxLib
- Large data files (" plotfiles") written at certain time steps; checkpoint files too.
- I/O consumes a significant fraction of run time



I/O R/W use case: Nyx/Boxlib

- Able to achieve max BW to single BB node by tweaking # MPI writers, transfer size.
- BB performance scales up as you increase # BB nodes in allocation
- BB performance matches Lustre…



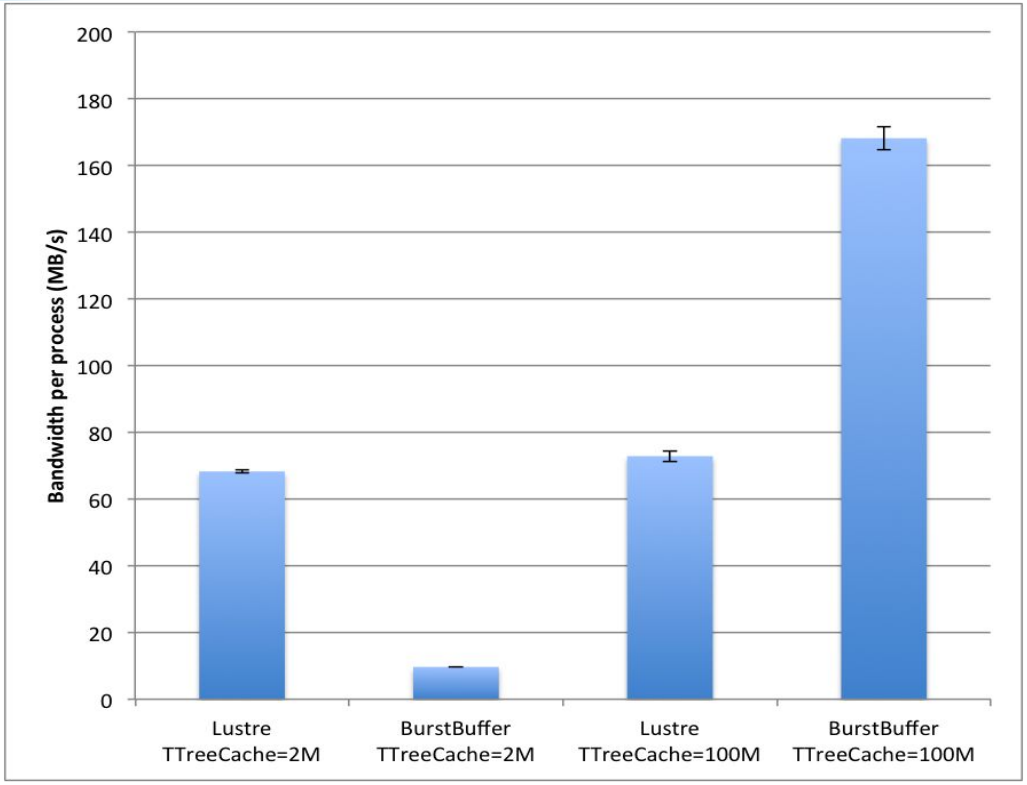
- Note that this does not necessarily correspond to optimal Nyx compute configuration!

Challenging I/O patterns

- **Benchmarks show promising results**
  - 12.5M IOP/s!
- **Reality more complex**
- **Lack of client-side caching significantly impacts performance compared to Lustre for small transfer sizes**
- **Applications tuned to use larger transfer sizes etc. saw better performance**
  - *Make them more like checkpoint use case*
- **DVS client-side caching and metadata improvements will help (coming later this year from Cray)**

Challenging IO use case: ATLAS data

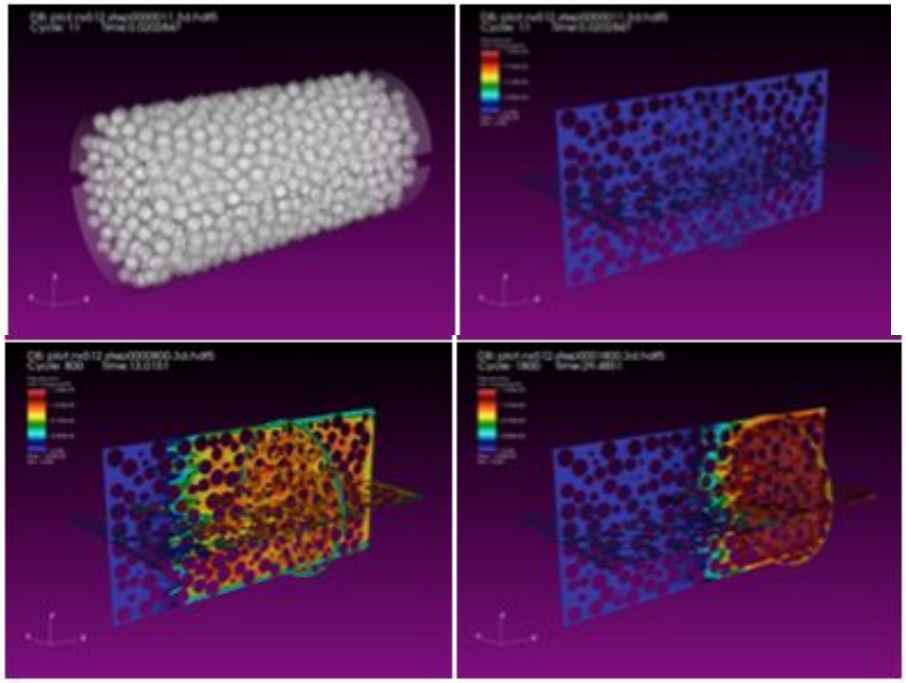
- **Initial study of I/O intensive data processing**
- **Reading 475 GB dataset in custom ROOT format**
- **32 forked processes per node, FPP R/W**
- **Initial result: BB performs poorly compared to Lustre.**



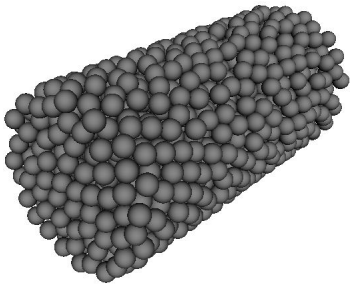
- **Increase application memory cache to 100 M**
  - Less reads – > 17x performance boost on BB

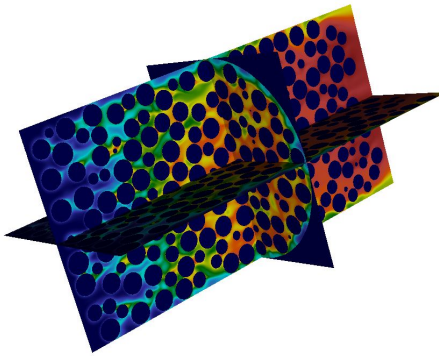
Workflow coupling and visualization

- **Success story: Burst Buffer can enable new workflows that were difficult to orchestrate using Lustre alone.**



Workflows Use Case: ChomboCrunch + VisIT

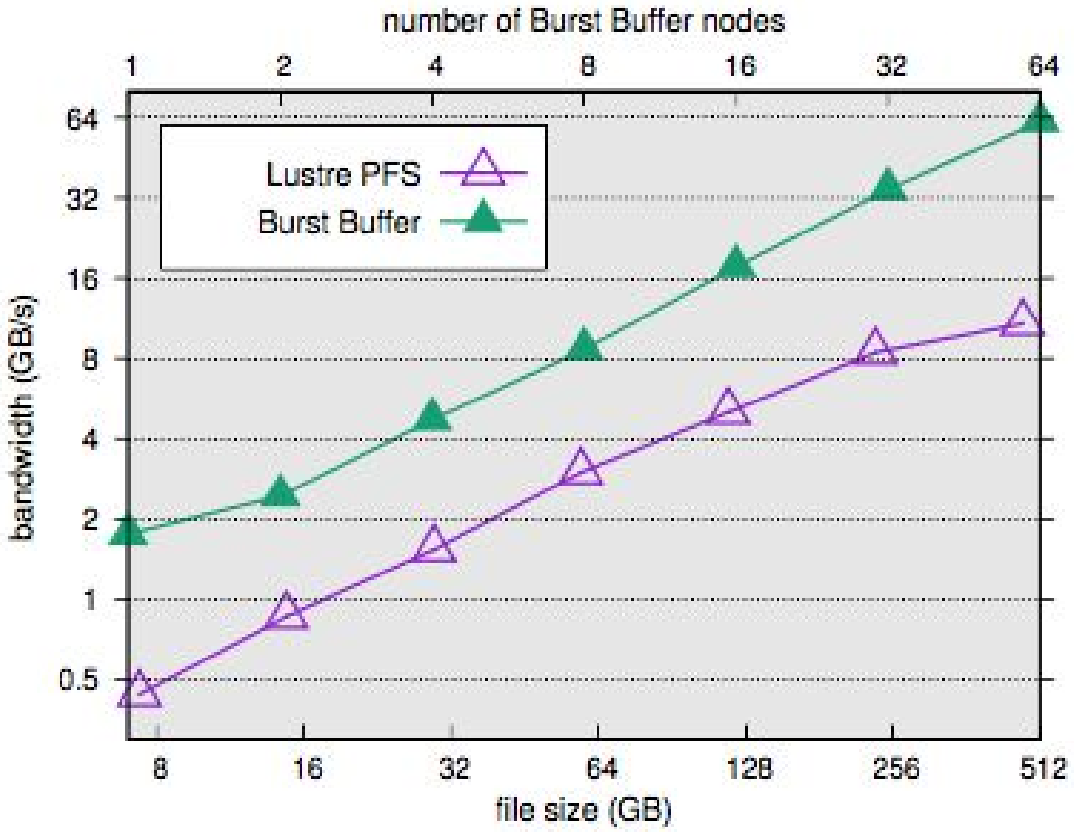




- **ChomboCrunch simulates pore-scale reactive transport processes associated with carbon sequestration**
  - Flow of liquids through ground layers
  - All MPI ranks write to single shared HDF5 '.plt' file.
  - Higher resolution -> more accurate simulation -> more data output (O(100TB))
- **VisIT – visualisation and analysis tool for scientific data**
  - Reads '.plt' files produces '.png' for encoding into movie
- **Before: used Lustre to *store* intermediate files.**

Scaling

- **Burst Buffer significantly out-performs Lustre for this application at all resolution levels**
  - Did not require any additional tuning!
- **Bandwidth achieved is around a quarter of peak, scales well.**



Compute node/BB node scaled: 16/1 to 1024/ 64
Lustre results used a 1MB stripe size and a stripe count of 72 OSTs

Summary: User Experience so far

- **Writing large files (with large block I/O) is fast (checkpointing use case)**
- **Reading/Writing small files (or small I/O transfers) is problematic in some cases**
  - Generally in many cases our BB performance is worse than our Lustre filesystem (which is high-performance).
  - Client-side caching helps Lustre performance
- **Still some system instabilities**
- **Initial enthusiasm from users somewhat diminished, but not extinguished!**

Expected performance improvements

1. **DVS client-side caching**
   - Lustre has client-side caching, currently DVS does not
   - Will help small R/W transfers on BB
   - Can currently use "iobuf" library for user-side caching
2. **Smaller granularity**
   - "Grain" is minimum amount of space allocated on each BB node, currently 213GB
     - E.g. request 500GB BB allocation - get 3x213GB "grains".
3. **MPI-IO shared file performance**
   - Change underlying file striping on BB (i.e. >3)

We're working with Cray to improve BB performance

- **Coming soon: transparent caching!**

Why not open to all users?

- **We've not yet accepted the Burst Buffer**
  - Still unstable, regular crashes
- **We're expecting big changes in how the Burst Buffer works with Rhine Redwood**
  - Transparent caching may affect current functionality
- **Only starting to work out a user policy**
  - Maximum allocation per user? Limit length of persistent reservation? How to schedule the resource?
- **Cori Phase 2 will also impact Burst Buffer usage**
- ***The Burst Buffer will not be generally available for some time***
  - But please contact us if you have a good use-case and we may be able to give you early access

Conclusions

- **Cori has a fully functional Burst Buffer**
  - Not yet open to all users…
- **Users are enthusiastic about new memory hierarchy!**
- **Burst Buffer has demonstrable utility beyond checkpoint/restart use case**
- **Very promising IO accelerator, but early stage of development**
  - Benchmarks good, user experience mixed…
- **Early User program excellent debugger of new hardware!**

CUG 2016 Best Paper!

More Burst Buffer info:

CUG 2016 Best Paper award:

https://www.nersc.gov/news-publications/nersc-news/nersc-center-news/2016/cug-honors-nersc-burst-buffer-early-user-program-with-best-paper/

More information:

http://www.nersc.gov/users/computational-systems/cori/burst-buffer/

Thankyou

Use Cases by BB feature

Application	I/O bandwidth: reads	I/O bandwidth: writes (checkpointing)	High IOPs	Workflow coupling	In-situ / in-transit analysis and visualization	Staging intermediate files/ pre-loading data
Nyx/Boxlib		X		X	X	
Phoenix 3D		X		X		X
Chomo/Crunch + Visit		X		X	X	
Sigma/UniFam/Sipros	X	X	X			X
XGC1	X	X				X
PSANA				X	X	X
ALICE	X					
Tractor			X	X		X
VPIC/IO					X	X
YODA			X			X
ALS SPOT/TomoPy	X			X	X	X
kitware				X	X	

Use Cases by BB feature

Application	I/O bandwidth: reads	I/O bandwidth: writes (checkpointing)	High IOPs	Workflow coupling	In-situ / in-transit analysis and visualization	Staging intermediate files/ pre-loading data
Electron cryo-microscopy						X
htslib						X
Falcon	X	X				
Ray/HipMer	X	X	X			X
CESM	X	X				
ACME/UV-CDAT					X	X
GVR		X				
XRootD				X		X
OpenSpeedShop	X	X				
DL-POLY		X				
CP2K		X				
ATLAS	X		X			X

National Energy Research Scientific Computing Center

New software versions set to default on Feb 28

- **pmi/5.0.10-1.0000.11050.0.0.ari** (from 5.0.9-1.0000.10911.0.0.ari)
- **cray-ga/5.3.0.5** (from 5.3.0.3)
- **cray-hdf5, cray-hdf5-parallel/1.8.16** (from 1.8.14)
- **cray-libsci/13.3.0** (from 13.2.0)
- **cray-mpich/7.3.1** (from 7.2.5)
- **cray-shmem/7.3.1** (from 7.2.5)
- **craype/2.5.1** (from 2.4.2)
- **fftw/3.3.4.6** (from 3.3.4.5)
- **papi/5.4.1.3** (from 5.4.1.2)
- **perftools, perftools-base, perftools-lite/6.3.1** (from 6.3.0)
- **cce/8.4.3** (from 8.4.0)
- **gcc/5.2.0** (from 5.1.0)